REMARKS

Claims 1-13 are pending in the application. Claims 1 and 11 have been amended herein. Favorable reconsideration of the application, as amended, is respectfully requested.

I. AMENDMENTS TO CLAIMS

Applicants have amended claims 1 and 11 herein to emphasize with even more specificity the features of the present invention. In view of the long pendency of the present application, applicants are hopeful such amendments will help expedite favorable prosecution.

Specifically, claims 1 and 11 have been amended to emphasize the particular structure of the hardware description language construct of the present invention. More particularly, claims 1 and 11 now recite that the language construct, for a given communication, *defines a sender process for sending the communication*. In addition, the language construct *defines a plurality of receiver processes each for receiving the communication sent by the sender process*. In turn, the language construct effects a synchronised communication between the sender process *and* the receiver processes.

Support for such amendment is found, for example, in the present application at page 9, line 10, to page 10, line 17. No new matter has been added.

For reasons discussed more fully below, the prior art neither teaches nor suggests a hardware description language construct as found in the present invention. More specifically, the prior art neither teaches nor suggests a language construct which, for a given communication, defines a sender process for sending the communication and a plurality of receiver processes each for receiving the communication sent by the sender process in synchronization.

II. REJECTION OF CLAIMS 1-10 BASED ON OBVIOUSNESS-TYPE DOUBLE PATENTING

Claims 1-10 remain rejected under the judicially created doctrine of obviousness-type double patenting in view of *USP 6,021,266 to Kay*. Withdrawal of the rejection is respectfully requested for at least the following reasons.

As noted above, claim 1 has been amended to recite a hardware language construct which, for a given communication, defines a sender process for sending the communication and defines a plurality of receiver processes each for receiving the communication sent by the sender process, in that synchronised communication between the sender process and the receiver processes is effected.

In *Kay*, the claims do not recite or suggest in any way a hardware language construct which defines a sender process and a plurality of receiver processes that each receive the communication from the sender process in synchronized communication therebetween. The claims do suggest a language construct which supports parallelism and synchronized communication (see, e.g., claim 1). For reasons described more fully below, however, the parallelism and synchronized communication referred to in *Kay* do not refer to a language construct defining a sender process and a plurality of receiver processes which each receive the communication from the sender process and thereby effect synchronized communication between the sender process and the receiver processes.

Rather, the parallelism and synchronized communication referred to in *Kay* relate to individual communications carried out through respective individual channels between a sender process and different receiver processes as explained more fully below. There is no definition of a sender process and plurality of receiver processes for effecting synchronized communication between the sender process and plurality of receiver processes as recited in claim 1.

Accordingly, withdrawal of the rejection with respect to claim 1 and remaining dependent claims 2-10 is respectfully requested.

III. REJECTION OF CLAIMS 1-13 UNDER 35 USC §102(b)

Claims 1-13 remain rejected under 35 USC §102(b) based on *Kay '245*. Withdrawal of this rejection is respectfully requested for at least the following reasons.

i. Claims 1 and 11

Claims 1 and 11, as amended, emphasize a language construct that, for a given communication, *defines a sender process for sending the communication*. In addition, the language construct *defines a plurality of receiver processes each for receiving the communication sent by the sender process*. In turn, the language construct effects a synchronised communication between the sender process *and* the plurality of receiver process*es*.

An example of the language construct referred to in claims 1 and 11 can be found at page 9, line 15 to page 10, line 7, reproduced below:

```
mchan int m1;      // declare m1 as an 'int' type
                   //          multichannel
par {{             // begin process branching
        …          // code for process 1
        send (m1, x);       // send value of x on m1, and synch
        …          // more code for process 1
    }{
        …          // code for process 2
        y = receive (m1);         // synchronise on m1,
                                        // store value in y
        …          // more code for process 2
}{
        …          // code for process 3
        z = receive (m1);         // synchronise on m1,
                                  // store value in z
        …          // more code for process 3
    }}             // end process branching
```

This example shows the declaration of a multichannel m1, and three processes which use it. The sender process ("send") sends the result of any expression E to m1

with the phrase send (m1, E);. In this case there are a plurality of receivers made up of two receivers x and y, and they may obtain the value sent by using the expression receive (m1). The send and receive constructs will not terminate until the entire synchronisation is complete; that is, when the sender has executed send and all receivers x and y have executed receive.

As exemplified above, the language construct defines, for a given communication, a sender process ("send") and a *plurality* of receiver processes (x and y) which communicate synchronously.

ii. Prior Art

In order for the Examiner to better appreciate the invention, applicants offer the following explanation of the prior art and *Kay '245*. In the background section of the present application, the computer language Occam is discussed. In Occam, the language construct may be used to define synchronous channels with each channel connecting exactly two processes (e.g., a sender process and a receiver process) synchronously. However, the communication is always between precisely two processes, namely a single sender process and a single receiver process. While parallel synchronous communications may occur along respective synchronous channels, Occam does not teach or suggest a synchronous communication channel which includes a sender process and a *plurality* of receiver processes. (See, e.g., Spec., p. 1, ln. 11 to p. 2, ln. 22). Thus, Occam does not teach or suggest a language construct that, for a given communication, defines a sender process for sending the communication and defines a plurality of receiver processes each for receiving the communication sent by the sender process, thereby effecting synchronised communication between the sender process and the receiver processes as recited in amended claims 1 and 11.

VHDL is a hardware description language that allows a single sender process to communicate with a plurality of receiver processes. However, the language construct itself provides only *unsynchronized* communication. (See, e.g., Spec., p. 2, ln. 24 to p. 3, ln. 14). The language construct in VHDL does not, for a given communication, define

a sender process for sending the communication and define a plurality of receiver processes each for receiving the communication sent by the sender process, thereby effecting synchronised communication between the sender process and the receiver processes as recited in amended claim 1 and 11.

Bach is a language similar to Occam in that parallel channels may be effected with synchronous communication. Again, however, each communication is between one sender process and one receiver process. There is no language construct which defines, for a given communication, a sender process and a plurality of receiver processes which receive the communication in synchronization. (See, e.g., Spec., p. 4, ln. 1 to p. 6, ln. 2).

iii. Kay '245

Regarding applicants' previous arguments that *Kay '245* does not teach or suggest "one-to-many" type of send-receive communications as recited in claims 1 and 11, the Examiner argues that such "one-to-many" send-receive communications would *inherently* result based on the operation described in *Kay '245*. (See, e.g., Office Action, page 14). Applicants respectfully disagree for at least the following reasons.

Kay '245 does not teach or suggest such "one-to-many" type of send-receive communication as recited in present claims 1 and 11. Rather, *Kay '245* teaches only a "one-to-one" type of send-receive communication. Each send is matched by only one receive. There is no construct for effecting synchronized communication between a sender process and a plurality of receiver processes.

The following example presented in *Kay '245* illustrates the manner in which the language construct does <u>not</u> define a sender process and a plurality of receiver processes for synchronously communicating as recited in amended claims 1 and 11:

```
chan int ch;
par {
        {
                inti = 0;
                while (1)
                        send (ch, i++);
```

```
}
{
        int tot = 0;
        while (1)
                tot += receive(ch);
}
```

The above example from *Kay '245* (p. 18) shows a pair of processes, a sender process which generates integers and sends them to the one receiver process (using the channel ch) which totals them. The language construct does not, for a given communication, define a sender process for sending the communication and define a plurality of receiver processes each for receiving the communication sent by the sender process in synchronization as recited in amended claims 1 and 11.

As further evidence that *Kay '245* does not teach or suggest a language construct as claimed, applicants note that *Kay '245* teaches that the language construct described therein is similar to that used in Occam. (p. 19, last paragraph). As noted above, Occam defines only point-to-point, or single sender to single receiver synchronous communication. While parallel communications may be established, they are always between an individual sender process and a corresponding individual receiver process (i.e., across different respective channels). There is no language construct which itself, for a given communication, defines a sender process for sending the communication and defines a plurality of receiver processes each for receiving the communication sent by the sender process, thereby effecting synchronised communication between the sender process and the receiver processes as recited in amended claim 1 and 11.

Thus, the features recited in claims 1 and 11 are neither expressly taught nor inherent in *Kay '245*.

Moreover, the present invention would not have been obvious in view of *Kay '245*. The Occam model upon which the teachings of *Kay '245* were based simply does not allow the one sender to many receiver synchronous communication. It was only by virtue of the present invention that all the processes are on the same device (the one designed using the hardware description language) that led to developments in

accordance with the present invention that permitted the one sender to many receiver feature.

Furthermore, applicants incorporate herein the distinctions set forth in their previous response, which equally apply now.

For at least the above reasons, withdrawal of the rejection of claims 1-13 is respectfully requested.

IV. CONCLUSION

Accordingly, all claims 1-13 are believed to be allowable and the application is believed to be in condition for allowance. A prompt action to such end is earnestly solicited.

Should the Examiner feel that a telephone interview would be helpful to facilitate favorable prosecution of the above-identified application, the Examiner is invited to contact the undersigned at the telephone number provided below.

Should a petition for an extension of time be necessary for the timely reply to the outstanding Office Action (or if such a petition has been made and an additional extension is necessary), petition is hereby made and the Commissioner is authorized to charge any fees (including additional claim fees) to Deposit Account No. 18-0988.

Respectfully submitted,

RENNER, OTTO, BOISSELLE & SKLAR, LLP

Mark D. Saralino
Reg. No. 34,243

DATE: April 19, 2004

The Keith Building
1621 Euclid Avenue
Nineteenth Floor
Cleveland, Ohio 44115
(216) 621-1113

C:\GEN\YAMA\yamap713.rceamd.wpd